$ AF
2124

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Mericas**

Serial No.: **09/310,912**

Filed: **May 13, 1999**

For: **Method and System for Counting Non-Speculative Events in a Speculative Processor**

§ Group Art Unit: **2124**
§
§ Examiner: **Chavis, John Q.**
§
§ Attorney Docket No.: **AT9-99-073**
§

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on August 5, 2002.

By: Amelia C. Nearing

TRANSMITTAL DOCUMENT

Assistant Commissioner of Patents
Washington, D.C. 20231

Sir:
ENCLOSED HEREWITH:

- Appellant's Brief (in triplicate) (37 C.F.R. 1.192); and
- Our return postcard.

A fee of $320.00 is required for filing an Appellant's Brief. Please charge this fee to IBM Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to Deposit Account No. 09-0447.

Respectfully submitted,

Duke W. Yee
Registration No. 34,285
CARSTENS, YEE & CAHOON, LLP
P.O. Box 802334
Dallas, Texas 75380
(972) 367-2001
ATTORNEY FOR APPLICANT

Docket No. AT9-99-073 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Mericas** §
§
Serial No.: **09/310,912** §
§
Filed: **May 13, 1999** §
§
For: **Method and System for Counting Non-Speculative Events in a Speculative Processor** §

§
Group Art Unit: **2124**
§
Examiner: **Chavis, John Q.**

#8
D. Cha
8-21-02

Assistant Commissioner for Patents
Washington, D.C. 20231

ATTENTION: Board of Patent Appeals and Interferences

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on August 5, 2002.

By: Amelia C. Nearing
Amelia C. Nearing

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on June 4, 2002.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

REAL PARTIES IN INTEREST

The real party in interest in this appeal is the following party: IBM Corporation

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-17

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: NONE
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1-17
4. Claims allowed: NONE
5. Claims rejected: 1-17

C. CLAIMS ON APPEAL

The claims on appeal are: 1-17

STATUS OF AMENDMENTS

With respect to amendments after final rejection, there are no such amendments.

SUMMARY OF INVENTION

The present invention provides a performance monitor for a speculative processor. See specification, page 10, lines 8-28. The processor allows instructions to execute out-of-order and employs speculative execution to predict the outcome of conditional branches of certain instructions before the data on which the certain instructions depend is available. The performance monitor includes performance monitor counters. See specification, page 8, lines 18-26. The performance monitor may include one or more interim counters that hold the count of occurrences for monitored events that are caused by instructions that may never complete. See specification, page 16, line 19, to page 17, line 7. When an instruction completes, the contents of the completed instruction's interim counter is added to a performance monitor counter. See page 17, lines 8-23. The present invention may also compute a difference between performance monitor counters and interim counts to generate a count of events related to speculatively executed instructions. See page 20, line 29, to page 21, line 6.

ISSUES

The issues on appeal are as follows:

Whether claims 1, 3, 4, 6-10, 12, and 14-17 are unpatentable as being anticipated by Dollin et al. (US Patent No. 6,112,236); and,

Whether claims 2, 5, 11, and 13 are unpatentable as being obvious over Dollin et al. (US Patent No. 6,112,236).

GROUPING OF CLAIMS

The claims on appeal do not stand or fall in a single group, but are grouped into in the following groups:

Claims 1, 12, and 16 form group A. Claim 4 forms group B. Claims 6, 15, and 17 form group C. Claim 7 forms group D. Claims 3 and 14 form group E. Claim 8 forms group F. Claim 9 forms group G. Claim 10 forms group H. Claims 2, 5, 11, and 13 form group I.

ARGUMENT

A. 35 U.S.C. § 102, Anticipation

The Office Action rejects claims 1, 3-4, 6-10, 12 and 17 under 35 U.S.C. § 102(e) as being anticipated by Dollin et al. (U.S. Patent No. 6,112,236 and *Dollin* hereinafter). This rejection is respectfully traversed.

The Office Action states:

Claims	Dollin
1. A method of monitoring events within a data processing system comprising a speculative processor and a plurality of counters, wherein each counter among said plurality of counts occurrences of specified events within the data processing system, the method comprising:	see title, abstract and figure 5, Item 59 (counts -- plurality of Counters). The processor is considered inherent to a system for monitoring events.
specifying an event to be monitored;	See col. 5 lines 63-67 and col. 7 lines 30-33 (predetermined implies Prior Specifying of an event). Also, See col. 8 lines 2-6.
monitoring for the specified event during the execution of instructions by the speculative processor;	see col. 15 lines 17-20.
generating a count of occurrences of the specified event for all instructions executed by the speculative processor; and	see the inserted events in figure 6B.
generating a count of occurrences of the specified event for instructions completely executed by the speculative processor.	see the corrupted events, col. 5 lines 37-40, which are utilized to help Determine the quality of service, see Col. 5 lines 46-60.

Office Action dated August 28, 2001 page 2. Appellant respectfully disagrees. *Dollin* teaches a method and apparatus for making quality of service measurements on a connection across a

network. The apparatus of *Dollin* tracks events in existing traffic carried by a connection of interest. See col. 3, lines 16-19. In other words, the apparatus and method of *Dollin* monitors for events in data units transmitted across the connection of interest. *Dollin* teaches generating counts of lost, inserted, and corrupted events for the connection. See col. 4, line 48, to col. 5, line 39.

In contradistinction, the present invention monitors for events that occur during the execution of instructions by a speculative processor, generates a count of occurrences of the events for all instructions executed by the speculative processor, and generates a count of occurrences of the events for instructions completed by the speculative processor. A speculative processor is a modern processor that may speculatively execute instructions that may be canceled or flushed without completely executing because the condition for which they were speculatively executed did not occur. See specification, page 3, lines 1-6. By generating separate counts for events that occur for all instructions executed by the speculative processor and events that occur for instructions completed by the speculative processor, a count of events that may be related to speculatively executed instructions may be determined.

The Office Action states, "[t]he processor is considered inherent to a system for monitoring events." The Office Action misapplies the concept of "inherent" anticipation. Section 102 of Title 35 deals with novelty and loss of patent rights. An invention is said to be "anticipated" when it is squarely described or disclosed in a single reference as identified from one of the categories of 35 U.S.C. § 102, commonly referred to as "prior art". Express anticipation occurs when the invention is expressly disclosed in the prior art, patent or publication. In some cases, however, when the claimed invention is not described *in haec verba*, the "doctrine of inherency" is relied on to establish anticipation. Under the principles of inherency, a claim is anticipated if a structure in the prior art necessarily functions in accordance with the limitations of a process or method claim. *In re King*, 801 F.2d 1324, 231 U.S.P.Q. 136 (Fed. Cir. 1986). A prior art reference that discloses all of a patent's claim limitations anticipates that claim even though the reference does not expressly disclose the "inventive concept" or desirable property the patentee discovered. *Verdgaal Brothers, Inc. v. Union Oil Company of California*, 814 F.2d 628, 2 U.S.P.Q.2d 1051, (Fed. Cir. 1987). It suffices that the prior art process inherently possessed at that property. *Id.* Mere possibilities or even probabilities, however, are not enough to establish inherency. The missing claimed characteristics must be a

"natural result" flowing from what is disclosed. *Continental Can Co. v. Monsant Co.*, 948 F.2d 1264, 20 U.S.P.Q.2d 1746 (Fed. Cir. 1991). Unstated elements in a reference are inherent when they exist as a "matter of scientific fact". *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 7 U.S.P.Q.2d 1057 (Fed. Cir.), *cert. denied*, 488 U.S. 892 (1988) and *Hughes Aircraft Co. v. United States*, 8 U.S.P.Q.2d 1580 (Ct. Cl. 1988). Otherwise, the invention is not inherently anticipated.

In the present case, *Dollin* teaches local processing units **12**, **14** at each end of the connection of interest. However, *Dollin* does not teach a "speculative processor" as recited in the claims. The Office Action provides no analysis as to why any processor of *Dollin* is, necessarily, a speculative processor. The examiner's assertion that these elements are present can be made only through the use of the appellants' disclosure as a template to fill in the missing elements.

The Final Office Action states:

> The applicant appears to claim (on page 3 of 8) that Dollin does not utilize a processor. However, see col. 8 lines 36-45 and see again fig. 1. He goes on to describe a specific type of processor (speculative processor), which "may speculatively execute instructions that may be canceled or flushed without completely executing because the condition for which they were speculatively executing did not occur". The definition provided is the essence of event processing. Therefore, if the event occurs something happens and if it does not occur the selected action does not occur. Those features were discussed in the previous action and again indicated by the applicant in his response and therefore will not be repeated here. Furthermore, events can occur in one end of the network or both (since each have processors to detect the occurrence of a specified event), see col. 7 line 23-col. 8 line 63.

Final Office Action, dated April 23, 2002. Appellant respectfully disagrees. Appellant clearly pointed out that *Dollin* teaches a pair of processors. Therefore, the Final Office Action mischaracterizes the statement made by appellant.

Furthermore, the logic used in the Final Office action is fatally flawed. If something happens in response to an event occurring, then there is no speculation. The event processing proposed by the examiner does not speculatively execute instructions that may be canceled or flushed without completely executing because the condition for which they were speculatively executing did not occur, because the event processing proposed by the examiner never speculatively processes any instructions. As stated in the Final Office Action, an action is either taken or it is not taken. This is an example of conditional processing, not speculative processing.

Even assuming, *arguendo*, that *Dollin* teaches a speculative processor, *Dollin* does not teach "monitoring for the specified event **during the execution of instructions by the speculative processor**," "generating a count of occurrences of the specified event for **all instructions executed by the speculative processor**," or "generating a count of occurrences of the specified event for **instructions completely executed by the speculative processor**," as specifically recited in combination, in claim 1. The events in *Dollin* are associated with data units transmitted over a network connection. *Dollin* does not teach or suggest associating events with instructions executed by a speculative processor or instructions completely executed by a speculative processor.

The Final Office Action also states:

> In reference to applicant's assertion that Dollin does not teach "monitoring for the specified event during the execution of instructions by the speculative processor", see again the abstract of the invention which indicates that "An event is deemed to have occurred at a monitored network point (one side of the network – ie one processor) when a predetermined set of criteria (instruction) concerning one or more data units is satisfied..." The applicant should also note that the mere monitoring of events is the execution of instructions. The previous listing in the abstract clearly indicates that monitoring of events occur **during execution of instructions by the speculative processor**" based on specified criteria, see also col. 8 lines 2-5. Also, in reference to the count means, Dollin keeps a count of discarded events (events that did not execute completely – ie Flushed events) and matched events (**instructions completely executed**) and generates reports based on the quality of service (inherently including all events), col. 5 lines 4-46.

Final Office Action, dated April 23, 2002. Appellant respectfully disagrees. *Dollin* makes no association between lost, inserted, and corrupted events for a network connection and **instructions executed by a speculative processor** or **instructions completely executed by a speculative processor**, because *Dollin* has nothing to do with speculative processors. The Final Office Action is performing calisthenics with the reference, bending and stretching teachings in the reference that have nothing to do with the present invention and warping them into something *Dollin* did not contemplate. For example, the Final Office Action refers to "discarded" events as "Flushed events." However, flushing an instruction from a processor is very different from dropping a packet in a network connection. The Final Office Action ignores the spirit of the invention, as well as the actual claim language, in an attempt to make an otherwise inapplicable reference allegedly applicable.

Moreover, the Office Action proffers no analysis as to why the "inserted events" of *Dollin* are equivalent to occurrences of the specified event for instructions completely executed by the speculative processor. Furthermore, the Office Action proffers no analysis as to why the "corrupted events" of *Dollin* are equivalent to occurrences of the specified event for instructions completely executed by the speculative processor. Clearly, this is because *Dollin* is not concerned whatsoever with a speculative processor.

Since *Dollin* does not teach each and every claim limitation, the claims are not anticipated by the applied reference. Therefore the rejection of claim 1 is overcome. Independent claims 4, 6, 7, 12, 15, 16, and 17 recite similar features to those addressed above with respect to claim 1 and are allowable for the same reasons. Additionally, claims 4, 6, 7, 12, 15, 16, and 17 recite other additional combinations of features not suggested by the reference.

Particularly, claim 4 recites "associating an interim counter with a particular instruction," "associating a first global event counter with all instructions," "associating a second global event counter with completed instructions," and "in response to detecting a completion of the particular instruction, adding event counts from the interim counter to the second global event counter." Since *Dollin* does not teach or suggest these features, claim 4 cannot be anticipated by *Dollin*.

With respect to claim 4, the Final Office Action states:

> In reference to applicant's global events (claim 4), see Dollin's quality of service feature which generates reports based on "all instructions", "completed instructions" and "lost or failed instructions", as known in the art, col. 5 lines 47-55 and col. 12 lines 3-8.

Final Office Action, dated April 23, 2002. Appellant respectfully disagrees. The quality of service feature of *Dollin* makes no association between lost, inserted, and corrupted events for a network connection and instructions executed by a speculative processor. The quoted passages are nowhere found in *Dollin*. The Final Office Action is misrepresenting the teachings to make it appear that the reference anticipates features that are simply not taught. For example, lost events in a network connection are not "lost or failed instructions," particularly as recited in claim 1.

Also, claim 6 recites "computing a difference between the count of occurrences of the specified event for all instructions and the count of occurrences of the specified event for all completed instructions as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor." The Office Action does not address this

feature. Since *Dollin* does not teach or suggest this feature, claim 6 cannot be anticipated by *Dollin*.

With respect to claim 6, the Final Office Action states:

> As per claim 6, the features are considered the essence of reports generated, see again the rejection of claim 4.

Final Office Action, dated April 23, 2002. Appellant respectfully disagrees. *Dollin* fails to teach speculative processors; therefore, any reports in *Dollin* would have nothing to do with "computing a difference between the count of occurrences of the specified event for all instructions and the count of occurrences of the specified event for all completed instructions as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor," as recited in claim 6.

Further, claim 7 recites "in response to detecting an occurrence of a particular specified event, incrementing a first counter and a second counter" and "in response to detecting a completion of an instruction, adding the second counter to a third counter." The Office Action does not address this limitation. Since *Dollin* does not teach or suggest, this feature, claim 7 cannot be anticipated by *Dollin*.

With respect to claim 7, the Final Office Action states:

> In reference to claim 7, note in the abstract that "whenever an event is detected, an event report is generated". Therefore, Dollin is considered to utilize a first counter (whenever an event is detected, see abstract and the (Quality of Service) QOS unit in col. 11 lines 61-col. 12 line 8) and a second counter (to indicate when a match does not occurs) and the third counter for keeping count of the completed matches to enable efficient report generation and specifying QOS as is known in the art, as a percentage of matches completed over the total.

Final Office Action, dated April 23, 2002. Appellant respectfully disagrees. Surely, the *Dollin* system does a great job of reporting the quality of service of a network connection. However, this is completely irrelevant to the present invention, which recites incrementing a first counter and a second counter when a specified event occurs and adding the second counter to a third counter when an instruction is completed.

Since claims 3, 8-10, and 14 depend from claims 1, 7, and 12 the same distinctions between *Dollin* and the invention recited in claims 1, 7, and 12 apply for these claims. Additionally, claims 3, 8-10 and 14 claim other additional combinations of features not suggested

by the reference. For example, claim 3 recites "monitoring a plurality of specified events for each instruction executed by the speculative processor." The Office Action alleges that this limitation is taught by *Dollin* in col. 5, lines 3-40 and col. 7, lines 30-33. These lengthy portions of *Dollin* teach a summary of the invention and "event types." However, nowhere does *Dollin* teach monitoring a plurality of specified events **for each instruction executed by the speculative processor**. The Office Action proffers no analysis as to why "event types" are equivalent to monitoring events for each instruction executed by a speculative processor. Since *Dollin* does not teach or suggest this feature, claim 3 cannot be anticipated by *Dollin*. Consequently, it is respectfully urged that the rejection of claims 1, 3-4, 6-10, 12 and 14-17 have been overcome.

In addition, Claims 8-10 are dependent upon claim 7. Claim 8 recites, "wherein the first counter counts occurrences of a particular specified event for all instructions executed by the speculative processor"; claim 9 recites, "wherein the second counter counts occurrences of a particular specified event for a particular instruction; and, claim 10 recites, "wherein the third counter counts occurrences of a particular specified event for instructions completely executed by the speculative processor." *Dollin* does not teach or suggest these features. The Office Action does not address these claims; therefore, the burden remains on the Office to establish a *prima facie* case of anticipation. As such, appellant asserts that claims 8-10 are allowable over *Dollin*.

Furthermore, *Dollin* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. *Dollin* actually teaches away from the presently claimed invention because it teaches monitoring events in data units transmitted across a network connection, as opposed to monitoring events that occur during execution of instructions by a speculative processor, as in the presently claimed invention. Absent the Office Action pointing out some teaching or incentive to implement *Dollin* for monitoring events within a data processing system comprising a speculative processor, one of ordinary skill in art would not be led to modify *Dollin* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Dollin* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the appellant's disclosure as a template to make the necessary changes to reach the claimed invention.

II. 35 U.S.C. § 103, Obviousness

The Office Action rejects claims 2, 5, 11 and 13 under 35 U.S.C. § 103(a) as being unpatentable over *Dollin*. This rejection is respectfully traversed.

The Office Action states:

2. The method of 1 further comprising: computing a difference between the count of occurrences of the specified event for all completed instructions as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor.	this feature is not specifically mentioned by Dollin et al.; however, it would have been obvious to a person of ordinary skill in the art at the time of the invention to compute the difference between inserted and Corrupted values enable the System to determine the Quality of Service being provided and to generate statistics of various measurements to enable report. Generation, see col. 5 lines 47-60.

Office Action dated August 28, 2001 page 4. Appellant respectfully disagrees. *Dollin* does not teach generating a count of occurrences of the specified event for all instructions or a count of occurrences of the specified event for all completed instructions. Therefore, it could not have been obvious to compute a difference between the count of occurrences of the specified event for all instructions and the count of occurrences of the specified event for all completed instructions.

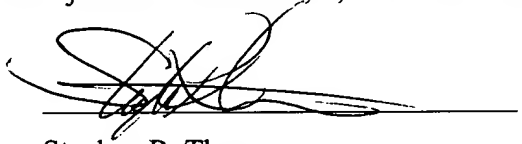
The Office Action alleges that it would have been obvious to compute a difference between inserted and corrupted values. Whether or not it would have been obvious to compute a difference between inserted and corrupted values is irrelevant, because the claims do not recite computing a difference between inserted and corrupted values. Furthermore, *Dollin* may suggest a motivation of determining quality of service measurements on a connection across a network. However, *Dollin* is completely unrelated to speculative processors. Therefore, the statistics and reports generated in *Dollin* would not lead a person of ordinary skill in the art to determine a count of occurrences of a specified event for instructions speculatively executed by a speculative processor.

The present invention recognizes the problem of monitoring events for instructions speculatively executed by a speculative processor. *Dollin* does not teach the problem or its source. Instead, *Dollin* is directed towards monitoring events for making quality of service measurements on a connection across a network. Therefore, one of ordinary skill in the art would not be motivated to combine or modify the references in the manner required to form the solution disclosed in the claimed invention.

In addition, the Office Action may not make modifications to the prior art using the claimed invention as a model for the modifications. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780, 1783-1784 (Fed. Cir. 1992). "The mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art has suggested the desirability of the modification." *Id.* In other words, unless some teaching exists in the prior art for the suggested modification, merely asserting that such a modification would be obvious to one of ordinary skill in the art is improper and cannot be used to meet the burden of establishing a *prima facie* cases of obviousness. Such reliance is an impermissible use of hindsight with the benefit of appellant's disclosure.

Therefore, absent some teaching, suggestion, or incentive in the prior art, *Dollin* cannot be properly modified to form the claimed invention. As a result, absent any teaching, suggestion, or incentive from the prior art to make the proposed modifications, the presently claimed invention can be reached only through the an impermissible use of hindsight with the benefit of appellant's invention as a model.

Claims 5, 11, and 13 recite similar features to those addressed above with respect to claim 2 and are allowable for the same reasons. Therefore, the rejection of claims 2, 5, 11 and 13 under 35 U.S.C. § 103 is overcome.



Stephen R. Tkacs
Reg. No. 46,430
Carstens, Yee & Cahoon, LLP
PO Box 802334
Dallas, TX 75380
(972) 367-2001

APPENDIX OF CLAIMS

The text of the claims involved in the appeal is:

1. A method of monitoring events within a data processing system comprising a speculative processor and a plurality of counters, wherein each counter among said plurality of counters counts occurrences of specified events within the data processing system, the method comprising:

specifying an event to be monitored;

monitoring for the specified event during the execution of instructions by the speculative processor;

generating a count of occurrences of the specified event for all instructions executed by the speculative processor; and

generating a count of occurrences of the specified event for instructions completely executed by the speculative processor.

2. The method of 1 further comprising:

computing a difference between the count of occurrences of the specified event for all instructions and the count of occurrences of the specified event for all completed instructions as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor.

3. The method of 1 further comprising:

monitoring a plurality of specified events for each instruction executed by the speculative processor.

4. A method of monitoring events within a data processing system comprising a speculative processor and a plurality of counters, wherein each counter among said plurality of counters counts occurrences of specified events within the data processing system, the method comprising:

associating an interim counter with a particular instruction;

associating a first global event counter with all instructions;

associating a second global event counter with completed instructions;

specifying an event to be monitored;

monitoring for the specified event during execution of instructions by the speculative processor;

in response to detecting an occurrence of the event during execution of the particular instruction, incrementing the interim counter;

in response to detecting an occurrence of the event during execution of any instruction, incrementing the first global counter; and

in response to detecting a completion of the particular instruction, adding event counts from the interim counter to the second global event counter.

5. The method of 4 further comprising:

computing a difference between event counts from the first global event counter and event counts from the second global event counter as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor.

6. A method of computing a count of events within a data processing system comprising a speculative processor and a plurality of counters, wherein each counter among said plurality of counters counts occurrences of specified events within the data processing system, the method comprising:

reading from a first counter a count of occurrences of a specified event for all instructions executed by the speculative processor;

reading from a second counter a count of occurrences of the specified event for instructions completely executed by the speculative processor; and

computing a difference between the count of occurrences of the specified event for all instructions and the count of occurrences of the specified event for all completed instructions as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor.

7. A method of monitoring events within a data processing system comprising a speculative processor and a plurality of counters, wherein each counter among said plurality of counters counts occurrences of specified events within the data processing system, the method comprising:

specifying events to be monitored;

monitoring said specified events;

in response to detecting an occurrence of a particular specified event, incrementing a first counter and a second counter; and

in response to detecting a completion of an instruction, adding the second counter to a third counter.

8. The method of 7 wherein the first counter counts occurrences of a particular specified event for all instructions executed by the speculative processor.

9. The method of 7 wherein the second counter counts occurrences of a particular specified event for a particular instruction.

10. The method of 7 wherein the third counter counts occurrences of a particular specified event for instructions completely executed by the speculative processor.

11. The method of 7 further comprising:

computing a difference between the first counter and the third counter to generate a count of occurrences of a particular specified event for speculatively executed instructions.

12. An apparatus for monitoring events within a data processing system comprising a speculative processor and a plurality of counters, wherein each counter among said plurality of counters counts occurrences of specified events within the data processing system, the apparatus comprising:

means for specifying an event to be monitored;

means for monitoring for the specified event during the execution of instructions by the speculative processor;

first means for generating a count of occurrences of the specified event for all instructions executed by the speculative processor; and

second means for generating a count of occurrences of the specified event for instructions completely executed by the speculative processor.

13. The apparatus of 12 further comprising:

means for computing a difference between the count of occurrences of the specified event for all instructions and the count of occurrences of the specified event for all completed instructions as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor.

14. The method of 12 further comprising:

means for monitoring a plurality of specified events for each instruction executed by the speculative processor.

15. A system for computing a count of events within a speculative processor comprising a plurality of counters, wherein each counter among said plurality of counters counts occurrences of specified events within the speculative processor, the system comprising:

first means for reading from a first counter a count of occurrences of a specified event for all instructions executed by the speculative processor;

second means for reading from a second counter a count of occurrences of the specified event for instructions completely executed by the speculative processor; and

means for computing a difference between the count of occurrences of the specified event for all instructions and the count of occurrences of the specified event for all completed instructions as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor.

16. A computer program product on a computer-readable medium for monitoring events within a data processing system comprising a speculative processor and a plurality of counters, wherein each counter among said plurality of counters counts occurrences of specified events within the data processing system, the computer-program product comprising:

first instructions for specifying an event to be monitored;

second instructions for monitoring for the specified event during the execution of instructions by the speculative processor;

third instructions for generating a count of occurrences of the specified event for all instructions executed by the speculative processor; and

fourth instructions for generating a count of occurrences of the specified event for instructions completely executed by the speculative processor.

17. A computer program product on a computer-readable medium for computing a count of events within a speculative processor comprising a plurality of counters, wherein each counter among said plurality of counters counts occurrences of specified events within the speculative processor, the computer program product comprising:

first instructions for reading from a first counter a count of occurrences of a specified event for all instructions executed by the speculative processor;

second instructions for reading from a second counter a count of occurrences of the specified event for instructions completely executed by the speculative processor; and

third instructions for computing a difference between the count of occurrences of the specified event for all instructions and the count of occurrences of the specified event for all completed instructions as a count of occurrences of the specified event for instructions speculatively executed by the speculative processor.